International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 9 (2015) pp. 23263-23279

http://www.ripublication.com

Advanced Eclat Algorithm for Frequent Itemsets Generation

Manjit Kaur, Urvashi Garg, Sarbjit Kaur

Email:kaurmanjit188@gmail.com,urvashi.16983@lpu.co.in,sainisarbjit27@gmail.com
Computer Science and Engineering,
Lovely Professional University
Phagwara, Punjab, India.

Abstract

Frequent itemset calculation plays crucial role in most of the organization. Frequent itemsets are those items which are frequently occurring in the database. A Lot of algorithms have already been designed for generating frequent itemsets. One of the algorithms that we have used is Eclat algorithm. Eclat algorithm uses vertical dataset and bottom up approach for searching items in database. But eclat algorithm has some limitations. For example, large number of iterations is required for processing the items and more escape time is required for finding frequent itemsets. Eclat algorithm uses bottom up approach that is very complex. We have improved eclat algorithm to reduce escape time and number of iterations by using top down approach and transposing the original data base.

Keywords- Frequent Itemset Mining, Eclat Algorithm, Advance eclat algorithm, Transposed Database Algorithm.

I Introduction

Data mining is the process of extracting useful information from various sources. These sources may be any relational database, transactional database, data warehouse etc. There are huge amount of data generated in the various organizations. Therefore organizer has to take number of decisions during extraction of useful data from the huge amount of data. But it is difficult to extract each and every record. So organizer finds frequently occurring data in the database. Based on those frequent data, organizers takes decision and makes business strategy. Frequent items are also very essential part in the online markets. In online markets, wide variety of products is available for customers. But customer gets confused while choosing among those products. So the products which are mostly purchased by various customers

(frequently purchased products) help new customer for best product selection. Different algorithms have been developed for calculation of frequent itemsets. Apriori is a very basic algorithm. But it takes a lot of time for calculations of frequent itemsets. In this algorithm, we need to calculate support and confidence. We need to scan to database again and again. So FP growth algorithm has been developed. FP growth stands for frequent pattern growth. In this algorithm, database scanning is required two times. Time consumption is more. To remove these limitations, Eclat algorithm was developed. Eclat algorithm is very commonly used algorithm. Eclat algorithm uses vertical database and depth first search approach. All the data is stored in vertical form. Bottom up approach is used for searching items in the database. Searching starts from bottom to top. Only support is calculated in this algorithm. But it takes more time than top down approach. There is no need to calculate confidence as it is not required and also because confidence increases complexity. Therefore this also not included in proposed algorithm. Firstly we need to calculate support of all items individually. We can decide support in two ways one which is decided by user and second by averaging all items. After calculating support of all items we will compare support of items with minimum decided support. The items which has support more than or equal to decided support are frequent itemsets. This is whole process of eclat algorithm. But this algorithm has limitations. Access time is more in this algorithm. Numbers of iterations are more. Complexity of eclat algorithm is more because of using bottom up approach. This algorithm can be improved by performing transpose operations. In transpose operations, all the items are given to all transactions at same time. It will save time also. Also top down approach is used in proposed algorithm. Number of iteration will be reduced by transposing database.

This paper is organized in 6 sections where section II describes Literature survey, section III explained Eclat algorithm, section IV gives description of Proposed Algorithm, section V explains Experimental Results and section VI provides Conclusion and future work in our research.

II Literature Review

Frequent itemsets play crucial role in organization and online markets. Many algorithms are proposed by different authors.

[1] **Jitendra Agrawal, Shikha Agrawal, Ankita Singhai· Sanjeev Sharm**a proposed SARIC(set particle swarm optimization for association rules using the itemset range and correlation coefficient). Apriori is basic algorithm for finding frequent itemsets but it takes more time for generating association rules in large database. Eclat is developed to remove limitations of apriori but eclat need user defined threshold. SARIC removes limitations of apiroi and eclat. It generates associations rules with less time.

[2] **Sandy Moens, Emin Aksehirli and Bart Goethals** works on frequent itemset mining for big data in which distributed version of eclat algorithm is defined. Frequent itemsets from big data is found. Big data is nothing but it contains huge amount of data. So finding frequent itemset on big data is very useful. Frequent item set mining plays very important role for extracting the useful

knowledge. In this paper, two algorithms are used one is Dist-Eclat algorithm which is used for speed purpose and second algorithm is BigFIM that focuses on extracting data from big databases. Hybrid approach is used with BigFIM for mining. In this paper frequent item set mining techniques are applied on the Map Reduce platform. Distributed version of eclat (Dist-eclat) divides the database into different processing units. Eclat concerns with speed based on K-FI. This k-FI is extracted through apriori algorithm and search frequent item sets using eclat. But it takes more processing time.

[3] **Marghny, H.Mohamed ,Mohammed,M.Darwieesh** works on efficient frequent itemsets algorithm. Frequent itemsets are very important for various data mining tasks and for generating association rules. In this paper, there are two algorithm are used. CountTableFI algorithm and BinaryCountTableF algorithm. Both these algorithms are different from apriori and other algorithms. The main idea for developing these algorithms is to present all transactions in binary form and decimal form. Because binary and decimal form of transactions can be easily understandable. By using binary and decimal form, user can use subset and identical set properties. In Countable algorithm, original transaction data is converted into new smaller transaction data. Then it generates new 'merge transactions' where' merge transaction' is collection of various transactions and then frequent itemsets are generated by creating count of table of items. In BinaryCountTableF algorithm, original transaction dataset is represented in 0/1 form. By using 0/1 form, user can convert data into decimal form. Then all transactions are merged. User can generate frequent itemsets using transactions.

[4] **Bina Kotiyal, Ankit Kumar, Bhaskar Pant, R.H. Gaudar,Shivali chahuan and Sonam Junee** work on user behaviour analysis in web log through comparative study of eclat and apriori . WWW provides all the needs of user on web. Web log files are generated on web. These web log files stores the information about interaction between client and service provider. It also stores information about web pages which are accessed by user. The information which keeps in the web log files is used to predict the behaviour of user. In this paper, two algorithms are used to determine which pages are accesses by user. These algorithms are eclat and apriori. Eclat is more scalable than apriori because in eclat algorithm fewer tables are generated as compared to apriori. So less time is required to perform the analysis in eclat algorithm.

[5] **Shaobo Shi.Yui Qi,Qin Wang** work FPGA acceleration for intersection computation in frequent itemset mining in which performance of eclat algorithm is increased because of huge amount of "sorted-set intersection computation" decrease the performance of algorithm. FPGA is one of the platforms which can be applied on parallel data.

[6] **Guo-Cheng Lan, Tzung-Pei Hong, Hong Yu Lee** introduced weighted frequent itemsets concepts. In Weighted frequent itemsets, weight values are set for all items. Weighted frequent itemsets mining improve data mining techniques. Transactions upper bound model is used to find upper bound. Upper bound is highest weighted value of items.

[7] **PENG JIan , WANG Xiao-ling** proposed an improved association rule algorithm that is based on itemset matrix and cluster matrix. Itemset matrix decreases time of comparison of candidate itemsets and records. Cluster itemsets makes cluster of records. When database is changed then only changed part is scanned. There is no need to scan full database again.

[8] **kan sin** works on new algorithm for discovering associations rules that is enlightened by eclat algorithm called LOGeclat. Logeclat find frequent pattern by using special candidate. Special candidate continue update data and reduce time. Logeclat is combination of eclat and special candidate method. It extracts correct itemsets. It takes less time then eclat but numbers of iterations are more.

[9] **M.N. Noothir** proposed Log server files which are very useful for each organization network. It develops large amount of data daily. The data contains useful and meaningful information. This information is very difficult to understand. So web log analysis tool is used for understanding information. In this paper Apriori algorithm is used to find frequent itemsets and UMPNA tool is used. UMPNA tool cleans the data. By using this tool, network performance is increased.

[10] **Sang Lin,Hu-yan Cui, Ren Ying, Zhou-lin Lin** say that Association rule mining helps for finding all association rules between all the itemsets in the database. Support and confidence must satisfy in the algorithm. There are numbers of constraints like knowledge constraints, data constraints but main constraint is item constraints. "B constraint" is major constraint. "B Constraint" is a predicate for power set of collection of itemsets. "B Constraint" can be divided into correspond classes. And before applying constraints on eclat algorithm, sorting of all items is done. When all constraints are applied then re-sort of all items is done. It is mainly used for dealing with others constraints. In this paper, only maximal itemset are found which satisfied all the constraints. It is very helpful in finding constraint frequent item sets in long pattern database.

[11] **Mahanti , Aniket and Reda Alhajj** proposed visual interface for displaying the result of eclat and apriori algorithm. Visual interface is very user friendly interface. User can easily understand it. By this interface, user can understand the frequent itemsets. User can judge that which item is mostly used. There is various application of association rule mining. Main application is market basket analysis. Market basket analysis is mostly required in market, malls, and online markets. There are other applications available such as customer profiling, fraud detection, credit risk analysis and so on.

III Eclat algorithm

Eclat algorithm finds the items from bottom like depth first search. Eclat algorithm is very simple algorithm to find the frequent item sets. This algorithm uses vertical database. Original database is converted into vertical database.

ALGORITHM:

Input: $F_k = \{I_1, I_2, ..., I_n\}$ // cluster of frequent k-itemsets.
Output: Frequent l-itemsets.
Bottom-Up (F_k) {
1. for all I_i € F_k
2. $F_k + 1 = \varnothing$;
3. for all I_j € F_k , i < j
4. $N = I_i \cap I_j$;
5. if N .sup≥= min_sup then
6. $F_k + 1 = F_k + 1 \cup N$;
7. end
8. end
9. end
10. if $F_k + 1 \neq \varnothing$ then
11. Bottom-Up ($F_k + 1$);
12. end
13.}

Take F_k as input. Output will be Set of frequent itemsets. In Bottom-Up (F_k), for loop is defined in which all items belong to database F_k under first step. In step2, take $F_k + 1$ as empty database. In step3, check that item exists in the database F_k or not. If item exists in the database F_k then calculate support of item in step4 where support means how many times item occurs in database F_k. In step5, compare support of items individually with minimum decided support. In step6, put those items whose support is more than then minimum support in database $F_k + 1$. In step7, check that database $F_k + 1$ is empty or not. If it is not empty then start the same procedure for other database.

.

Example of Eclat Algorithm:

Table 1: Original Database

Transaction	ITEM1	ITEM2	ITEM3	ITEM4	ITEM5
101	0	0	1	0	0
102	0	1	1	1	0
103	1	1	0	1	1
104	1	1	0	1	1
105	1	0	0	1	0
106	0	1	1	1	1

107	1	0	1	0	0
108	0	0	1	1	1

In table1, 8 transactions are taken. 5 items are sold in 101 to 108 transactions. 0 represent item does not exist in the database and 1 represent item exists in the database.Table2 shows support of each item in every transaction i.e. items comes in how many transactions and compare support of each item with decided support. Table3 shows only those items whose support is equal to and greater than decided support. In table4, we have paired items of table3 and calculate their support i.e. in total numbers of transactions in which their pairs comes collectively. Table5 shows only those paired items whose support is equal to and greater than decided support. In table6, we have paired items of table5 and calculate their support i.e in total numbers of transactions in which their pairs comes collectively. Table7 shows only those paired items whose support is equal to and greater than decided support.

Table 2: **Calculating Support**

Itemset	**Support**
ITEM1	4
ITEM2	4
ITEM3	5
ITEM4	6
ITEM5	4

Average support is 3. Take items which has support equal to or more than 3.

Table3: **Table after comparing Support**

Itemset	**Support**
ITEM1	4
ITEM2	4
ITEM3	5
ITEM4	6
ITEM5	4

Table4: Paired Database

Itemset	Support
ITEM1,ITEM2	2
ITEM1,ITEM3	1
ITEM1,ITEM4	1
ITEM1,ITEM5	2
ITEM2,ITEM3	2
ITEM2,ITEM4	4
ITEM2,ITEM5	3
ITEM3,ITEM4	3
ITEM3,ITEM5	2
ITEM4,ITEM5	4

Average support is 3.

Table5: **Paired Database after checking support**

Itemset	Support
ITEM2,ITEM4	4
ITEM2,ITEM5	3
ITEM3,ITEM4	3
ITEM4,ITEM5	4

Table6: Paired Database

Itemset	Support
ITEM2,ITEM4,ITEM5	3
ITEM2,ITEM3,ITEM4	2
ITEM3,ITEM4,ITEM5	2

Table7: Result Database

Itemset	Support
ITEM2,ITEM4,ITEM5	3

IV Proposed Algorithm

PROPOSED AGORITHM:

Input: Data set is in Transposed form contain transactions and items
Output: FI Frequent Itemset

1. Initialize: P =(T(i),T(j)) for all items in dataset T [Here T(i) is number of rows and T(j) is the number of columns in the dataset T]
2. [row column]=size(P) to get the number of rows and columns in the dataset P
3. While(P(I, j)=[rows columns])
4. for(i=0;i=P(rows); i++) [it is the for loop to count number of items]
5. for (j=0;j=P(columns);i++)
6. If (p(i , j)<=1)
7. count=count+1;
8. Else
9. count=count;
10. end
11. end
12. end
13. if (count>=support)
14. P(i,j)=P((i-1),P(j-1))U(P(i),P(j);
15. else
16. P(i,j)=P(i,j);
17. end
18. FI=P((i-1),P(j-1))U(P(i),P(j));
19. end

In this algorithm, numbers of transactions are taken as input. Output will be set of frequent itemsets. First Initialize P as one dataset which contains the information about rows and columns of transposed dataset. Then put all information of dataset P into [row column]. Then checks support of all transactions and compare it with minimum support. The transactions which satisfied minimum support are taken in FI database.

Example of Proposed Algorithm

In table8, take transpose of table1 (Original Database). Calculate support of each transaction in table9 and then compare support of each transaction with minimum support. Take those transactions in table10 which satisfy minimum support.

Make pair of transactions and calculate support of each pair in table11. Take all pairs in Table12 which satisfy decided support. Take three transactions as a pair in table13 and calculate support of each pair. Pairs which satisfy minimum support are taken in table14 (Final result database).

Table8: Transposed Database

ITEM	TID	TID	TID	TID	TID	TID
ITEM1	103	104	105	107		
ITEM2	102	103	104	106		
ITEM3	101	102	106	107	108	
ITEM4	102	103	104	105	106	108
ITEM5	103	104	106	108		

Table9: Support Database

TID	Support
101	1
102	3
103	4
104	4
105	2
106	4
107	2
108	3

Average support is 3. So we take those Transactions whose support is 3 or more than 3.

Table10: Support Database after support checking

TID	Support
102	3
103	4
104	4
106	4
108	3

Table11: Paired Database

TID	Support
102,103	2
102,104	2
102,106	3
102,108	2
103,104	3
103,106	3
103,108	2
104,106	3
104,108	2
106,108	3

Average support is 3. So we take those Transactions whose support is 3 or more than 3.

Table12: Paired Database after checking Support

TID	Support
102,106	3
103,104	3
103,106	3
104,106	3
106,108	3

Table13: Paired Database

TID	Support
102,103,106	2
102,104,106	2
102,106,108	2
103,104,106	3
103,106,108	2
104,106,108	2

Minimum support is 3. Take those transactions which are above or equal to minimum support.

Table14: Result Database

TID	Support
103,104,106	3

V Experimental Results

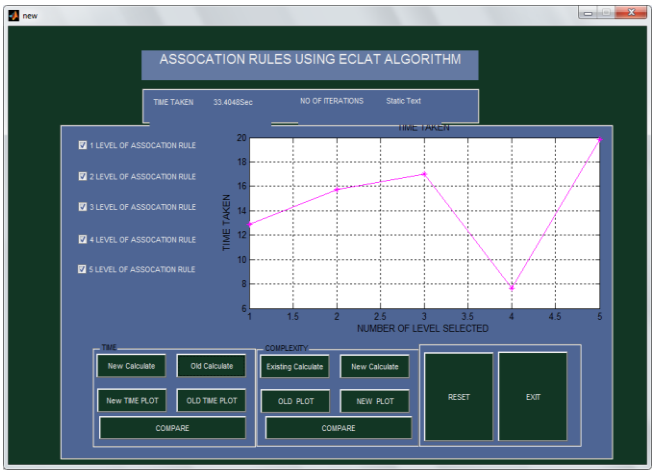

Fig.1.1: Time analysis of existing algorithm

As illustrated in figure 1.1, the various level of association rules have been shown, in this figure time had been calculated to generate association rules using basic eclat algorithm with vertical scan and graph is shown correspond to level of association rules.

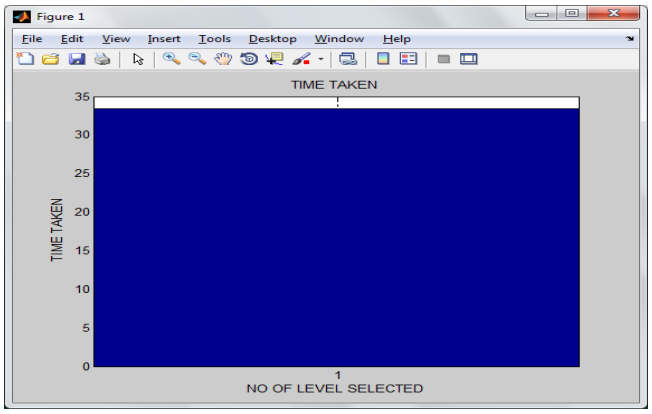

Fig 1.2 Individual Graph of time taking by eclat algorithm

As illustrated in figure 1.2, the eclat algorithm will be implement using bottom-up parsing and using vertical scan of database. In this bar graph is shown that how much algorithm will take to create association rules. This bar graph is created by clicking on old time plot button.

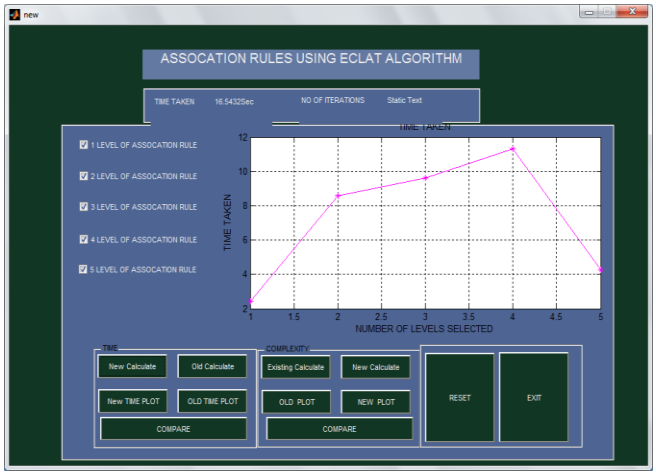

Fig: 1.3: Time taken by enhanced algorithm

As shown in figure 1.3, the enhancement will be proposed in eclat algorithm to reduce processing time of the algorithm. This will reduce the time to create processing to algorithm to create final association rules.

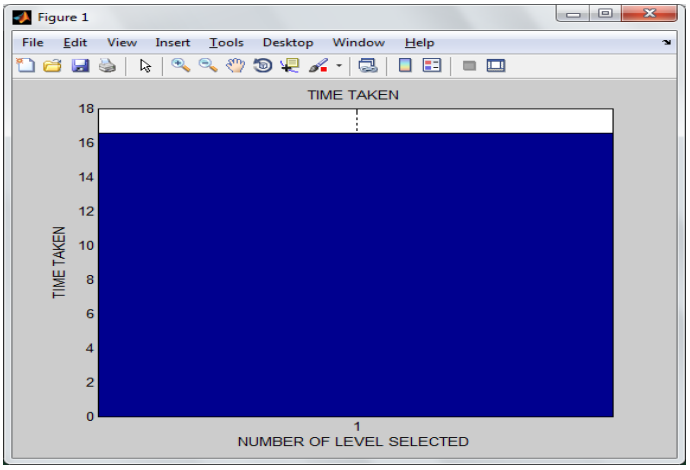

Fig 1.4: Enhanced algorithm Individual time graph

As shown in figure 1.4, the enhanced algorithm is proposed in which the transpose of the original dataset is taken and for data parsing top-down technique is implement which reduce processing time of the algorithm as shown in the bar graph.

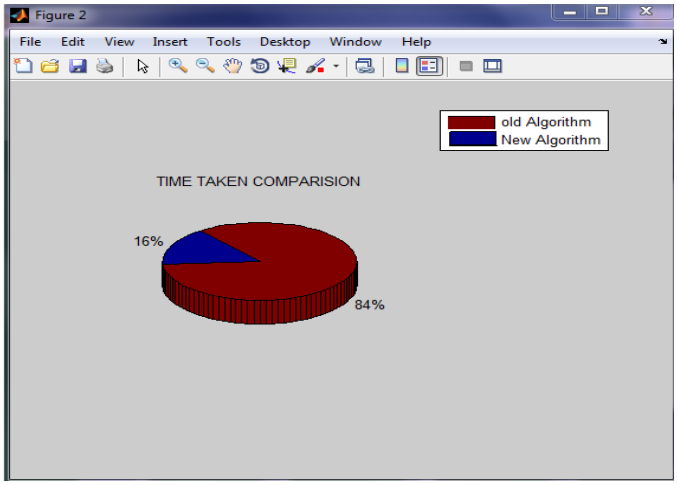

Fig.1.5 Time comparison

As shown in figure 1.5, the two algorithms had been implemented, the first algorithm is basic eclat algorithm in which vertical database and bottom-up technique is used for database scan. In second algorithm which is the enhancement of basic eclat algorithm in which transposed database is taken and top-down technique is implement which reduce the processing time as shown in the pie chart. The red portion shows the processing time of basic algorithm and blue portion shown the processing time of proposed algorithm.

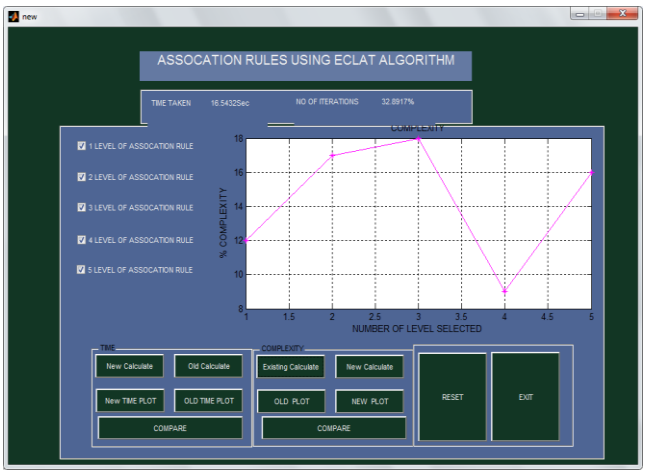

Fig.1.6 Complexity calculation of Basic Eclat Algorithm

As illustrated in figure 1.6, the various level of association rules have been shown, in this figure complexity had been calculated to generate association rules using basic eclat algorithm with vertical scan and graph is shown correspond to level of association rules.

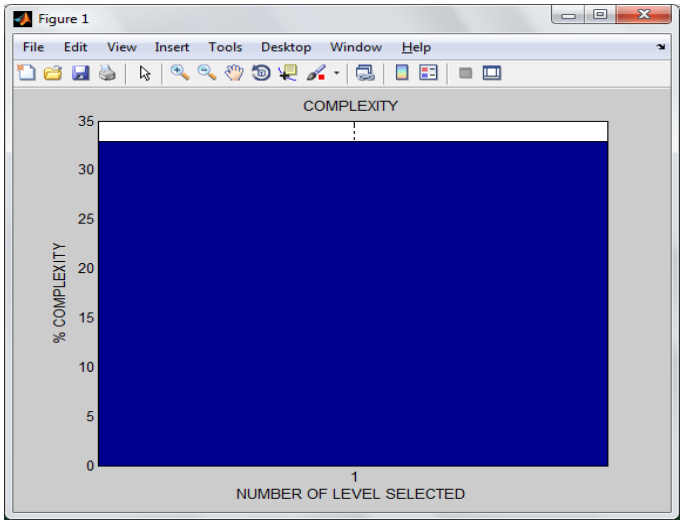

Fig 1.7: Individual Graph of Complexity of old algorithm

As illustrated in figure 1.7, the eclat algorithm will be implement using bottom-up parsing and using vertical scan of database. In this bar graph is shown that complexity of algorithm to create association rules.

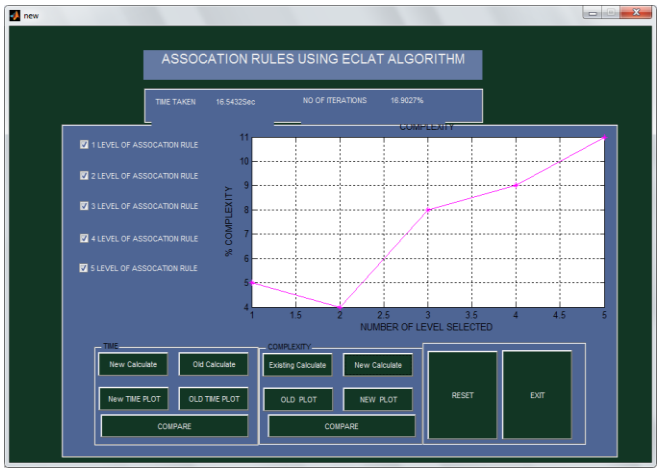

Fig.1.8 Complexity analysis for enhanced algorithm

As shown in figure 1.8, the enhancement will be proposed in eclat algorithm to reduce complexity of the algorithm. In enhanced algorithm, the horizontal scan will be proposed and top-down approach is implemented for parsing. This will reduce the complexity of proposed algorithm to create final association rules.

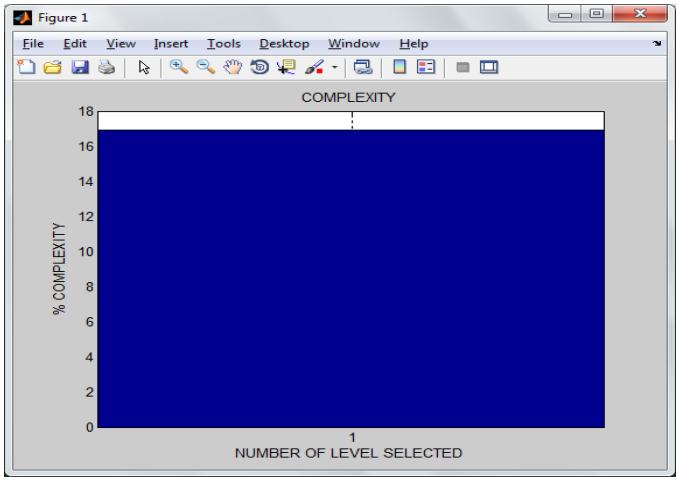

Fig 1.9: Individual Graph of Complexity of proposed algorithm

As shown in figure 1.9 the enhanced algorithm is proposed in which the transpose of the original dataset is taken and for data parsing top-down technique is implement which complexity of the algorithm as shown in the bar graph

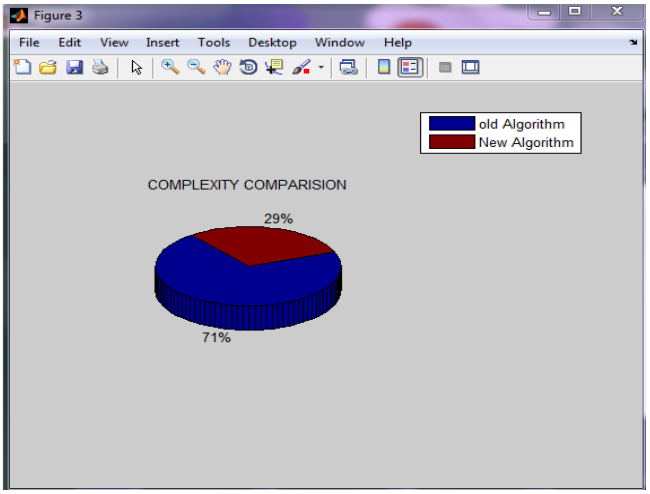

Fig. 1.10 Comparison Graph

As shown in figure 1.10, the two algorithms had been implemented, the first algorithm is basic eclat algorithm in which vertical database and bottom-up technique is used for database scan. In second algorithm which is the enhancement of basic eclat algorithm in which transposed database is taken and top-down technique is implement which reduce the complexity of algorithm as shown in the pie chart. The red portion shows the complexity of enhanced algorithm and blue portion shown the complexity of basic algorithm

VI Conclusions

Frequent itemsets play very important role in our day to day life. Eclat algorithm is used to find frequent itemsets. But many problems are found in eclat algorithm while finding frequent itemsets. Like Large number of iterations is required for processing the items from huge database and also more escape time and more complexity has been found in eclat algorithm. To remove these problems, advanced eclat algorithm is developed which is based on calculating support values only. Number of iterations and escape time is get decreased by using transposed database. In the advanced eclat algorithm, Top-Down Approach is used by which complexity is decreased. As shown in the experimental results, proposed algorithm has high scalability and good speedup ratio. In this research, comparison had been made with calculating support values which shows that proposed algorithm is more efficient in terms of processing time and complexity and this algorithm has provide best way to find products. In future, transposition of database will be applied on Apriori algorithm to analyse the performance in the term of escape time and number of iterations.

VII References

[1] SET-PSO-based approach for mining positive and negative association rules,Jitendra Agrawal, Shikha Agrawal, Ankita Singhai, Sanjeev Sharma, November 20140219-1377

[2] Moens, S.; Aksehirli, E.; Goethals, B., "Frequent Itemset Mining for Big Data," *Big Data, 2013 IEEE International Conference on* , vol., no., pp.111,118, 6-9 Oct. 2013

[3] Frequent itemsets algorithms**"** International Journal of Machine Learning and Cybernetics, 2013, Page 1 Marghny H. Mohamed, Mohammed M. Darwieesh

[4] Kotiyal, Bina; Kumar, Ankit; Pant, Bhaskar; Goudar, R.H.; Chauhan, Shivali; Junee, Sonam, "User behavior analysis in web log through comparative study of Eclat and Apriori," *Intelligent Systems and Control (ISCO), 2013 7th International Conference on* , vol., no., pp.421,426, 4-5 Jan. 2013

[5] Shaobo Shi; Yue Qi; Qin Wang, "FPGA Acceleration for Intersection Computation in Frequent Itemset Mining," *Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2013 International Conference on* , vol., no., pp.514,519, 10-12 Oct. 2013

[6] Guo-Cheng Lan; Tzung-Pei Hong; Hong Yu Lee; Shyue-Liang Wang; Chun-Wei Tsai, "Enhancing the Efficiency in Mining Weighted Frequent Itemsets," *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on* , vol., no., pp.1104,1108, 13-16 Oct. 2013doi: 10.1109/SMC.2013.192

[7] Peng Jian; Wang Xiao-ling, "An improved association rule algorithm based on Itemset Matrix and Cluster Matrix," *Computer Science & Education (ICCSE), 2012 7th International Conference on* , vol., no., pp.834,837, 14-17 July 2012

[8] Kan Jin, "A new algorithm for discovering association rules," *Logistics Systems and Intelligent Management, 2010 International Conference on* ,

vol.3, no., pp.1594,1599, 9-10 Jan. 2010

[9] Noorhuzaimi, M.N.; Junaida, S.; Mazrul, R.M., "An analysis of network services using association rules," *Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on* , vol., no., pp.469,473, 8-11 Aug. 2009

[10] Sang Lin; Hu-yan Cui; Ren Ying; Zhou-lin Lin, "Algorithm Research for Mining Maximal Frequent Itemsets Based on Item Constraints," *Information Science and Engineering (ISISE), 2009 Second International Symposium on* , vol., no., pp.629,633, 26-28 Dec. 2009

[11] Mahanti, Aniket, and Reda Alhajj. "Visual interface for online watching of frequent itemset generation in Apriori and Eclat." *Machine Learning and Applications, 2005. Proceedings. Fourth International Conference on*. IEEE, 2005.